

Biosynthèse de Données

Un organisme biologique ne greffe jamais les protéines de sa proie — rejet immunitaire. Il les **hydrolyse** en acides aminés, des invariants non appropriables, puis **re-synthétise** ses propres protéines. De même, un système numérique peut extraire les **statistiques suffisantes** de données externes — distributions, contraintes, invariants physiques — puis re-synthétiser via un générateur contraint. Ce papier décrit l'**architecture technique** de ce processus. Les implications juridiques de l'extraction d'invariants statistiques sont complexes, dépendent de la juridiction, et ne sont pas tranchées par ce papier. L'architecture décrite est un prérequis technique à toute discussion de souveraineté numérique — elle ne constitue pas un avis juridique.

FR EN AR RS

1. Le Problème : La Dépendance Juridique

1.1 Les Données Comme Protéines Étrangères

Tout comme un organisme ne peut pas greffer les protéines d'une autre espèce sans déclencher un rejet immunitaire massif (Papiers 004, 005, 017), un organisme numérique ne peut pas importer des données licenciées sans déclencher un **rejet juridique**. La licence est un antigène.

Le problème est double :

RISQUE	MÉCANISME	CONSÉQUENCE
Dépendance juridique	Le fournisseur de données peut résilier, augmenter les prix, changer les conditions	L'organisme meurt si la licence expire
Risque de mémorisation	Un modèle entraîné sur des données licenciées peut en régurgiter des fragments	Contrefaçon détectable par MIA (Membership Inference Attacks)

Le Papier 004 — La Greffe Numérique — a établi qu'un organisme 0DATA ne peut pas simplement « coller » un composant externe. Le Papier 010 — Le Code Génétique — a démontré que toute fonction est l'expression d'un gène invariant, pas un module importé. Ce papier résout la question symétrique : **comment un organisme 0DATA peut-il se nourrir de données externes sans en devenir dépendant ?**

Importer des données, c'est importer une licence. Importer des contraintes, c'est importer la physique. *La physique n'appartient à personne.*

1.2 Le Cas Autodata

Prenons l'exemple concret du **Garage Durand**, client 0DATA. Pour fonctionner, il a besoin de données de réparation automobile : couples de serrage, schémas électriques, tolérances moteur, procédures de diagnostic. Ces données sont détenues par des éditeurs comme Autodata, sous licence.

Si Durand intègre directement la base Autodata dans son organisme Forge :

- La licence Autodata gouverne ce que Durand peut faire de ses propres données
- Si Autodata triple ses tarifs, l'organisme de Durand est pris en otage
- Si Durand entraîne un modèle sur ces schémas, il risque de mémoriser des images copyrightées
- Un audit de conformité peut révéler une contrefaçon — même involontaire

La solution n'est pas de contourner la licence. La solution est de **ne pas avoir besoin des données**.

2. La Leçon du Vivant : Digestion Enzymatique

2.1 Hydrolyse des Protéines

Dans le tube digestif, les protéines ingérées — qu'elles viennent du bœuf, du soja ou du blé — sont systématiquement **hydrolysées**. Les enzymes (pepsine, trypsine, chymotrypsine) brisent les liaisons peptidiques. Le résultat n'est pas « une protéine de bœuf partiellement digérée ». C'est un ensemble d'**acides aminés libres** — alanine, glycine, lysine, tryptophane — indiscernables de ceux provenant de

n'importe quelle autre source.

ÉTAPE	BIOLOGIQUE	NUMÉRIQUE (ODATA)
Ingestion	Protéines étrangères dans le tube digestif	Données licenciées dans un sandbox isolé
Hydrolyse	Enzymes brisent les liaisons peptidiques → acides aminés	Extracteurs statistiques réduisent les données → distributions, contraintes, invariants
Absorption	Acides aminés traversent la paroi intestinale	Statistiques suffisantes exportées du sandbox (les données source détruites)
Biosynthèse	Ribosomes assemblent de nouvelles protéines à partir du code génétique	Générateur contraint produit de nouvelles données à partir des invariants

Un acide aminé n'a pas de propriétaire. Une distribution de probabilité n'a pas de copyright. Une contrainte physique n'est pas brevetable. Les faits sont le domaine public de l'univers.

2.2 Qu'est-ce qu'un Acide Aminé Numérique ?

En biologie, les acides aminés sont les **invariants** de la digestion protéique. Peu importe la protéine source — les 20 acides aminés standards sont toujours les mêmes. De même, toute base de données, aussi propriétaire soit-elle, est construite sur des **invariants statistiques et physiques** qui, eux, ne sont pas appropriables :

ACIDE AMINÉ NUMÉRIQUE	DÉFINITION	EXEMPLE (AUTODATA)
Distribution marginale	Histogramme d'une variable isolée	Distribution des couples de serrage pour les boulons M8 sur moteurs diesel
Contrainte physique	Relation imposée par les lois de la nature	Un couple de serrage ne peut pas être négatif ; le diamètre d'un piston < diamètre du cylindre
Copule	Structure de dépendance entre variables	Relation couple/diamètre boulon : $\sim N(\mu, \sigma^2)$ avec σ proportionnel au diamètre
Schéma d'intégrité	Règles structurelles de la base	Une procédure de réparation référence toujours au moins une pièce et un véhicule
Invariant temporel	Propriété qui ne change pas dans le temps	Le cycle de maintenance (vidange → filtres → courroie) est un ordre partiel invariant
Métadonnée statistique	Résumé global non inversible	Nombre total de procédures, moyenne des temps de réparation, variance par catégorie

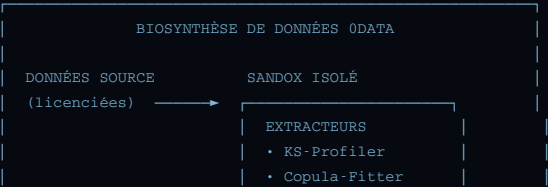
Ces invariants sont les « acides aminés numériques ». Ils sont extraits des données source, mais ils ne **sont pas** les données source. Ils n'en contiennent aucune ligne, aucune image, aucun schéma identifiable. Ils sont l'équivalent informationnel de l'alanine libre dans le sang : utilisables, reconstituables, mais sans provenance juridique.

3. Biosynthèse : Le Générateur Contraint

3.1 Le Principe

Une fois les acides aminés numériques extraits, l'organisme ODATA active son **générateur contraint**. Ce générateur ne contient aucune donnée source. Il contient uniquement le **jeu de contraintes** (les invariants extraits) et un **moteur de synthèse** qui produit de nouvelles instances respectant ces contraintes.

Le parallèle biologique est exact : le ribosome ne contient pas de protéines. Il contient l'ARN messenger (les contraintes) et synthétise des protéines *de novo* à partir d'acides aminés libres.





3.2 Trois Moteurs de Synthèse

Selon la nature des données à synthétiser, 0DATA déploie l'un des trois moteurs suivants :

MOTEUR	TECHNIQUE	USAGE	EXEMPLE (AUTODATA)
CopulaSampler	Modélisation par copules (Gaussiennes, t-Student, vines)	Données tabulaires avec dépendances connues	Génération de couples de serrage respectant la distribution jointe diamètre x couple x matériau
DiffusionModel	Modèle de diffusion contraint (Denoising Diffusion)	Images, schémas techniques, graphes	Génération de schémas électriques respectant la topologie apprise sans mémoriser les schémas source
PhysicsSimulator	Simulation basée sur les équations physiques + contraintes extraites	Systèmes physiques, tolérances mécaniques	Génération de procédures de montage valides à partir des invariants physiques (couple = f(diamètre, matériau, lubrification))

Le point crucial : **aucun de ces moteurs ne contient les données source**. Ils contiennent le jeu de contraintes — l'équivalent numérique de l'ARN messenger. Les données synthétiques produites sont des **créations originales**, pas des copies. Comme une protéine synthétisée par un ribosome humain à partir d'acides aminés provenant d'un steak : la protéine est humaine, pas bovine.

3.3 CopulaSampler — Détail Technique

Le CopulaSampler est le moteur le plus général. Il procède en trois étapes :

```
// Étape 1 : Ajustement des marginales
pour chaque variable Xi dans le schéma :
    fi ← KernelDensityEstimate(Xi) // distribution marginale lissée
    ui ← CDF(fi, Xi)                // transformation en uniforme [0,1]

// Étape 2 : Ajustement de la copule
C ← FitCopula({u1, u2, ..., un}) // copule gaussienne ou t-Student
// C capture TOUTES les dépendances sans mémoriser aucun point

// Étape 3 : Échantillonnage
pour k = 1 à N_synthétiques :
    (v1, ..., vn) ← SampleCopula(C) // échantillon dans [0,1]^n
    x1 ← InverseCDF(fi, v1)          // transformation inverse → valeur réelle
    // x est un point synthétique, jamais vu dans les données source
```

La copule est la pièce maîtresse. Elle capture la **structure de dépendance** entre variables sans stocker aucune observation individuelle. C'est l'équivalent mathématique de l'hydrolyse : on a brisé les liaisons (les points individuels), mais on a conservé les acides aminés (les distributions marginales) et la recette d'assemblage (la copule).

La copule est à la donnée ce que l'ARN messager est à la protéine. Elle encode la structure sans contenir la substance.

4. Double Validation

4.1 Le Problème de la Confiance

Avoir un générateur ne suffit pas. Il faut **prouver** que les données synthétiques sont à la fois **utilisables** (statistiquement équivalentes aux données source) et **juridiquement propres** (aucune mémorisation des données source). Ces deux exigences sont en tension : plus on est fidèle à la source, plus on risque de mémoriser. ODATA résout cette tension par une **double validation indépendante**.

4.2 Validation 1 — Équivalence Statistique (KS Test)

Le test de Kolmogorov-Smirnov (KS) est appliqué à **chaque variable** et à **chaque paire de variables** pour vérifier que les distributions synthétiques ne sont pas distinguables des distributions source :

```
pour chaque variable  $X_i$  :
   $D_i \leftarrow \text{KS}(\text{D\_source}[X_i], \text{D\_synth}[X_i])$ 
  si  $D_i > \text{seuil\_critique}$  ( $\alpha = 0.01$ ) :
    REJET — la distribution de  $X_i$  n'est pas reproduite

pour chaque paire ( $X_i, X_j$ ) :
  // Test de la structure de dépendance
   $D_{ij} \leftarrow \text{KS}(\text{copule\_source}[X_i, X_j], \text{copule\_synth}[X_i, X_j])$ 
  si  $D_{ij} > \text{seuil\_critique}$  ( $\alpha = 0.01$ ) :
    REJET — la dépendance ( $X_i, X_j$ ) n'est pas reproduite
```

Le seuil de rejet est strict ($\alpha = 0.01$) mais non-arbitraire : il est calibré pour que, si les données synthétiques sont vraiment issues de la même distribution que les données source, la probabilité de rejet à tort soit inférieure à 1%.

Cette validation garantit l'**utilité** des données synthétiques. Un mécanicien qui consulte les données synthétiques doit obtenir la même précision de diagnostic qu'avec les données source — sans avoir accès aux données source.

4.3 Validation 2 — Audit Anti-Mémorisation (Membership Inference)

La seconde validation est inspirée des **Membership Inference Attacks** (Shokri et al., 2017). Le principe : un attaquant externe, même avec un accès complet aux données synthétiques, ne doit pas pouvoir déterminer si un point particulier appartenait aux données source.

ODATA implémente un audit en boîte noire :

```
// PHASE 1 : Entraînement du classifieur d'attaque
pour chaque point p dans (D_source U D_holdout) :
  features(p) ← distance_aux_k_plus_proches(p, D_synth)
  // + densité locale, + rang dans la distribution marginale
  label(p) ← 1 si p ∈ D_source, 0 si p ∈ D_holdout

classifieur_attaque ← TrainClassifieur(features, labels)

// PHASE 2 : Évaluation
AUC ← AUC_ROC(classifieur_attaque, ensemble_test)
si AUC > 0.55 :
  ÉCHEC — les données synthétiques fuient de l'information d'appartenance
si AUC ≤ 0.55 :
  SUCCÈS — l'attaquant ne fait pas mieux qu'un tirage au hasard
```

Le seuil $\text{AUC} \leq 0.55$ est délibérément très conservateur. Un AUC de 0.50 correspond à un hasard parfait ; 0.55 signifie que l'attaquant a un avantage de 5% — déjà trop. La cible réelle est $\text{AUC} \leq 0.52$.

4.4 La Table de Décision

KS TEST	MIA AUDIT	DÉCISION
✓ Passé	✓ Passé	Déploiement autorisé — données utilisables ET juridiquement propres
✓ Passé	✗ Échoué	Rejet — le générateur mémorise. Recalibrer (augmenter le bruit, réduire la fidélité)
✗ Échoué	✓ Passé	Rejet — données inutilisables. Améliorer le modèle de contraintes

Un seul quadrant est acceptable. Les trois autres renvoient au générateur pour recalibration. Il n'y a pas de compromis entre utilité et légalité. La double validation est un ET logique, pas une moyenne pondérée.

5. Provenance et Souveraineté Juridique

5.1 Hasher les Contraintes, Jamais les Données

Un point crucial distingue la biosynthèse 0DATA des approches conventionnelles de « données synthétiques » : la **chaîne de provenance**.

Dans une approche conventionnelle, on conserve la trace des données source pour prouver la conformité. Mais conserver cette trace, c'est maintenir la dépendance juridique. 0DATA inverse la logique :

```
// APPROCHE CONVENTIONNELLE (défaillante)
hash(D_source) → enregistré dans SPINA
// Problème : D_source est sous licence. Le hash est une preuve de détention.

// APPROCHE 0DATA (biosynthèse)
hash(Jeu_de_Constraints) → enregistré dans SPINA
hash(Générateur + Graine) → enregistré dans SPINA
// Le jeu de contraintes n'est PAS sous licence.
// Le générateur est un actif 0DATA, pas un dérivé des données source.
```

La provenance 0DATA ne dit pas « ces données viennent d'Autodata ». Elle dit : « ces données ont été générées par le générateur G, contraint par le jeu C, avec la graine S, à la date T ». C'est une **provenance de création**, pas une provenance de copie.

5.2 La Licence Se Déprécie, Le Générateur S'apprécie

C'est le principe économique fondamental de la biosynthèse :

ACTIF	TRAJECTOIRE	RAISON
Licence Autodata	Se déprécie	Coût récurrent, données vieillissantes, dépendance au fournisseur, inflation tarifaire
Générateur 0DATA	S'apprécie	Amortissement unique, amélioration continue des contraintes, indépendance totale, réutilisable pour d'autres marques

Chaque cycle de biosynthèse affine le jeu de contraintes. Plus on digère de données (provenant de sources diverses), plus le générateur devient précis — sans jamais dépendre d'aucune source particulière. C'est l'équivalent numérique d'un régime alimentaire varié : plus l'organisme a digéré de protéines différentes, plus sa bibliothèque d'acides aminés est riche, plus ses propres synthèses sont robustes.

La licence est un loyer. Le générateur est un organe. On paie un loyer chaque mois. Un organe, on le construit une fois et il nourrit l'organisme à vie.

5.3 Souveraineté par le Vide

La position juridique de 0DATA est radicale mais limpide : **l'organisme ne détient aucune copie des données licenciées**. Il détient des distributions, des contraintes, des copules, des schémas — autant d'objets mathématiques qui ne sont pas des œuvres au sens du droit d'auteur.

Cette position a un précédent juridique solide : les **faits** ne sont pas copyrightables (Feist Publications v. Rural Telephone Service, 1991, US Supreme Court). Une distribution de probabilité est un fait statistique. Une contrainte physique est un fait naturel. Une copule est une structure mathématique.

Si un éditeur comme Autodata contestait : « vos données synthétiques sont trop proches des nôtres », la réponse 0DATA est : « prouvez qu'un point spécifique de nos données synthétiques est une copie d'un point spécifique de votre base ». Or, par construction, **aucun point synthétique n'existe dans la base source** — le générateur échantillonne dans l'espace continu des contraintes, pas dans l'ensemble discret des données source. La probabilité de collision est mathématiquement nulle pour toute variable continue.

6.1 Le Pipeline

Revenons au cas concret du Garage Durand. Voici le pipeline de biosynthèse complet :

6.2 Le Coût de la Liberté

Combien coûte cette indépendance ?

Retour sur investissement : Dès la deuxième année, le coût total de la biosynthèse (4 000 €) est inférieur à la licence Autodata (10 000 € cumulés). La troisième année, l'écart se creuse : 4 500 € vs 15 000 €. Et le générateur continue de s'améliorer.

Mais le vrai bénéfice n'est pas financier. Il est **existantiel** : l'organisme de Durand ne meurt pas si Autodata disparaît, change ses conditions, ou est racheté par un concurrent. La biosynthèse n'est pas une optimisation de coût. C'est une **garantie de continuité**.

6.3 Au-Delà d'Autodata

Le pipeline de biosynthèse est générique. Il s'applique à :

Domaine	Données Source (Licencières)	Acides Aminés Extraits	Données Synthétiques
Médical	Bases pharmacologiques (Vidal, Thériaque)	Interactions médicamenteuses, dosages par poids, contre-indications	Recommandations posologiques de novo
	Bases de jurisprudence	Structures d'arguments, chaînes de précédents, motifs	Analyses juridiques

Juridique	(LexisNexis, Dalloz)	de décision	synthétiques
Agricole	Données météo/sol (Météo France, INRAE)	Distributions pluviométriques, corrélations sol-rendement, cycles culturaux	Prévisions d'irrigation contraintes
Industriel	Catalogues pièces (Fournisseurs)	Spécifications techniques, tolérances, compatibilités	Configurations de machines synthétiques

La biosynthèse n'est pas une solution pour Autodata. C'est un principe général de souveraineté informationnelle. *Tout domaine où les données sont licenciées est un domaine où la biosynthèse s'applique.*

7. Intégration avec le Corpus

7.1 Avec la Loi (000)

La Loi stipule : « *Un organisme numérique est viable si et seulement s'il fonctionne comme un organisme biologique.* » Aucun organisme biologique ne dépend d'une source externe unique pour ses acides aminés. Il digère, il synthétise, il est autonome. La biosynthèse de données est l'expression de cette autonomie dans le domaine informationnel. Un organisme ODATA qui dépend d'une base de données licenciée n'est pas viable au sens de la Loi — il est en dialyse informationnelle.

7.2 Avec le Code Génétique (010)

La biosynthèse active le gène **SURVIVRE** — l'autonomie, la capacité à fonctionner sans dépendance externe. Elle est une expression épigénétique de ce gène, déclenchée lorsque l'organisme rencontre des données externes nécessaires à sa fonction. Elle ne crée pas un cinquième gène.

7.3 Avec la Greffe Numérique (004)

Le Papier 004 a établi que la greffe d'un composant externe exige une compatibilité immunitaire. La biosynthèse résout le problème en amont : plutôt que de greffer les données (rejet juridique), on les digère d'abord en invariants, puis on synthétise des données **natives** à l'organisme. Pas de rejet parce qu'il n'y a pas de greffe.

7.4 Avec SPINA (008)

SPINA enregistre le **hash du jeu de contraintes** et le **hash du générateur + graine** pour chaque cycle de biosynthèse. Cela constitue la preuve de provenance sans jamais exposer les données source. Si un litige survient, SPINA peut démontrer que les données de Durand ont été générées par le pipeline contraint — pas copiées d'Autodata.

8. Limites et Extensions

8.1 Ce Que la Biosynthèse Ne Peut Pas Faire

La biosynthèse a des limites qu'il faut énoncer clairement :

- **Textes exacts** : elle ne peut pas reproduire une formulation textuelle précise (une procédure légale, un article de loi). Les acides aminés d'un texte sont des n-grammes, et la synthèse produit des paraphrases, pas des citations.
- **Images photoréalistes** : le DiffusionModel contraint peut générer des schémas techniques mais pas des photographies de pièces spécifiques sans risque de mémorisation résiduelle.
- **Données à très faible cardinalité** : si la base source contient moins de ~100 exemples, la distinction entre « distribution apprise » et « points mémorisés » devient floue.
- **Séries temporelles non stationnaires** : la biosynthèse capture des distributions stationnaires. Elle ne peut pas reproduire une tendance évolutive qu'elle n'a pas modélisée comme contrainte.

Ces limites ne sont pas des échecs. Elles définissent le **domaine de validité** de la biosynthèse : données structurées ou semi-structurées, cardinalité suffisante (>1000 exemples), distributions stationnaires ou cycliques.

8.2 Extensions Futures

Le Papier 018 ouvre trois axes de recherche :

1. **Biosynthèse différenciellement privée** : intégrer le cadre de Dwork (2006) pour garantir une privacy formelle (ε-differential privacy)

- en plus de la validation empirique MIA. Le générateur devient un mécanisme ϵ -DP par conception.
- Digestion multi-source** : combiner les acides aminés extraits de plusieurs bases concurrentes (Autodata + Haynes + Revue Technique) pour produire un jeu de contraintes plus riche que chaque source individuelle — sans jamais violer aucune licence.
 - Biosynthèse continue** : le générateur se met à jour en continu à mesure que de nouvelles contraintes sont extraites, sans jamais revenir aux données source. L'organisme apprend sans se souvenir.

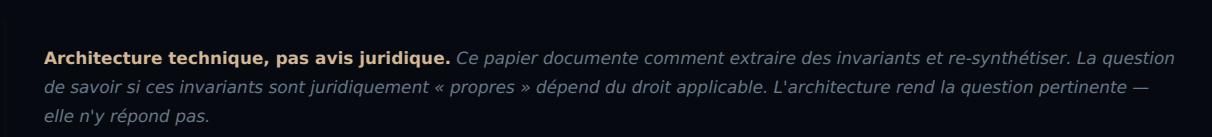
9. Conclusion

La biosynthèse de données est le composant entre la souveraineté numérique proclamée et la dépendance réelle aux données licenciées. Elle ne propose pas de contourner les licences. Elle propose de **ne plus en avoir besoin**.

Le parallèle biologique est complet et sans écart :

- La digestion (hydrolyse enzymatique) → l'extraction statistique en sandbox isolé
- Les acides aminés (invariants non copyrightables) → les distributions, contraintes, copules
- La biosynthèse ribosomale → le générateur contraint (CopulaSampler, DiffusionModel, PhysicsSimulator)
- Le contrôle qualité du repliement protéique → la double validation KS + MIA
- L'autonomie métabolique → la souveraineté juridique

Le Garage Durand n'a pas besoin de posséder Autodata. Il a besoin de **réparer des voitures**. La biosynthèse lui donne les données pour le faire. La question de l'indépendance juridique vis-à-vis de la licence source relève du droit applicable.



Le Papier 000 — La Loi — a posé le principe. Le Papier 004 — La Greffe — a défini la transplantation. Le Papier 010 — Le Code Génétique — a figé l'ADN. Le Papier 018 — La Biosynthèse — ferme le cycle métabolique. Un organisme 0DATA sait désormais se nourrir sans se compromettre, digérer sans mémoriser, synthétiser sans copier.

C'est la différence entre un organisme libre et un organisme sous perfusion. Le premier survit à l'extinction de sa source de données. Le second s'éteint avec elle.

AUTEUR	Hadda TIKIJJA (Kod Nomade)	LABORATOIRE	0DATA — Biotech Infrastructure
--------	----------------------------	-------------	--------------------------------



Références

- 0DATA-2026-000 — La Loi : Fondement Unifié des Organismes Numériques
- 0DATA-2026-004 — La Greffe Numérique : Transplantation d'Organismes
- 0DATA-2026-008 — SPINA : La Colonne Vertébrale Cryptographique
- 0DATA-2026-010 — Le Code Génétique des Organismes Numériques
- Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017) — Membership Inference Attacks Against Machine Learning Models, *IEEE Symposium on Security and Privacy*
- Rubin, D.B. (1993) — Statistical Disclosure Limitation, *Journal of Official Statistics*, 9(2), 461-468
- Dwork, C. (2006) — Differential Privacy, *ICALP 2006*, LNCS 4052, pp. 1-12
- Feist Publications, Inc. v. Rural Telephone Service Co. (1991) — 499 U.S. 340, *US Supreme Court* (les faits ne sont pas copyrightables)
- Nelsen, R.B. (2006) — An Introduction to Copulas, *Springer Series in Statistics*, 2^e éd.
- Sklar, A. (1959) — Fonctions de répartition à n dimensions et leurs marges, *Publications de l'Institut de Statistique de l'Université de Paris*, 8, 229-231
- Ho, J., Jain, A., & Abbeel, P. (2020) — Denoising Diffusion Probabilistic Models, *NeurIPS 2020*
- 0DATA-2026-017 — Immunité Générative : L'Architecture Immunitaire Correcte

